Identifying Structural Properties of Proteins from X-ray Free Electron Laser Diffraction Patterns

Paula Olaya
University of Tennessee
Knoxville, USA
polaya@vols.utk.edu

Silvina Caíno-Lores
University of Tennessee
Knoxville, USA
scainolo@utk.edu

Vanessa Lama
University of Tennessee
Knoxville, USA
vlama@vols.utk.edu

Ria Patel
University of Tennessee
Knoxville, USA
rpatel77@vols.utk.edu

Ariel Keller Rorabaugh
University of Tennessee
Knoxville, USA
aror@utk.edu

Osamu Miyashita
Center for Comp. Science, RIKEN
Kobe, Japan
osamu.miyashita@riken.jp

Florence Tama
Nagoya University & Center for Comp. Science, RIKEN
Nagoya & Kobe, Japan
florence.tama@nagoya-u.jp

Michela Taufer
University of Tennessee
Knoxville, USA
taufer@utk.edu

2022 IEEE 18th International Conference on e-Science (e-Science) | 978-1-6654-6124-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ESCIENCE55777.2022.00017

***Abstract*—Capturing structural information of a biological molecule is crucial to determine its function and understand its mechanics. X-ray Free Electron Lasers (XFEL) are an experimental method used to create diffraction patterns (images) that can reveal structural information. In this work we design, implement, and evaluate XPSI (X-ray Free Electron Laser-based Protein Structure Identifier), a framework capable of predicting three structural properties in molecules (i.e., orientation, conformation, and protein type) from their diffraction patterns. XPSI predicts these properties with high accuracy in challenging scenarios, such as recognizing orientations despite symmetries in diffraction patterns, distinguishing conformations even when they have similar structures, and identifying protein types under different noise conditions. Our framework shows low computational cost and high prediction accuracy compared to other machine learning methods such as random forest and neural networks.**

***Index Terms*—machine learning, diffraction patterns, proteins, autoencoder**

I. Introduction

Proteins and other biological molecules are responsible for many vital cellular functions, such as transport, signaling, or catalysis, and dysfunction can result in diseases. Information on the 3-dimensional (3D) structures of biological molecules and their dynamics is essential to understand mechanisms of their functions, leading to medicinal applications such as drug design. To obtain such information, a variety of experimental techniques have been developed. Since the 1950s, X-ray crystallography has been the primary technique to obtain structural information. Developments of X-ray free-electron laser (XFEL) light sources offer a new possibility for imaging biological systems. Its extremely strong X-ray laser allows imaging of biological systems without crystallization, and therefore, it can be applied to a wider variety of systems under various physiological conditions. In addition, such a strong light enables "single-shot" imaging, i.e., one can obtain a two dimensional (2D) diffraction image of "radiation damage-free" samples. If a sufficient number of these 2D diffraction patterns is collected it is then possible to generate a 3D structure, assuming the orientation of each diffraction pattern can be determined [1]–[3]. In addition, as bio-molecules are highly dynamic, XFEL may also capture the molecule in different conformations. In order to achieve high resolution structure determination, it is also critical to identify the conformation of the bio-molecules captured in a given diffraction image.

In this work we focus on identifying three structural properties of bio-molecules, orientation, conformation and protein type, embedded in 2D diffraction patterns using machine learning (ML) approaches. Orientation refers to the placement of the incident beam with respect to a protein structure and is defined by the three angles: Φ (Azimuth), Θ (Altitude), and Ψ (Rotation angle). Conformation determines the overall shape of the molecule. Protein type refers to the identity of the protein structure.

We present the design, implementation, and validation of a software framework for the predictions of the structural properties embedded in XFEL images. We call this framework XPSI (XFEL-based Protein Structure Identifier). XPSI is a framework that relies on ML methods such as an autoencoder and the K-nearest neighbor method (KNN), to capture key information that allows the identification of properties, such as spatial orientation, protein conformation, and different protein types from the diffraction patterns. Our framework comprises four key modules: (i) a data pre-processing module to load, crop, and normalize images; (ii) an autoencoder module to extract diffraction pattern features; (iii) a ML module to model and predict the three structural properties from the diffraction patterns; and (iv) a validation module to measure the accuracy of the predicted properties. We demonstrate the capabilities of our framework by evaluating and quantifying the accuracy when the framework is presented with challenging datasets and combinations of structural protein properties. Moreover, we provide a comparison analysis of our framework with other cutting-edge methods in two fronts: prediction accuracy and computational performance.

The contributions of this work are as follows:

- A framework to predict the three structural properties of protein (i.e., orientation, conformation, and protein type)

978-1-6654-6124-5/22/$31.00 ©2022 IEEE
DOI 10.1109/eScience55777.2022.00017

X-ray diffraction patterns;

- The validation of the prediction accuracy of XPSI for challenging scenarios with symmetrical orientations, conformational similarity, and protein identification under highly noisy conditions;
- The comparison of computational performance and prediction accuracy of our framework versus other cutting edge ML-based methods.

The paper is organized as follows, Section II describes the generation of diffraction patterns and the protein's structural properties. Section III presents our framework. Section IV demonstrates the capabilities of the framework for challenging scenarios with noise and symmetry in the diffraction patterns, high similarity of structural conformations, and multiple protein types. Section V quantifies the computational performance and prediction accuracy in comparison with other cutting-edge methods. Section VI summarizes findings and provides directions for future work.

II. Protein Diffraction Patterns and Properties

In this study, protein diffraction patterns obtained from XFEL experiments are considered. We provide a short description of the experimental generation of diffraction patterns and the data processing that is required to extract information embedded in these diffraction patterns.

A. Experimental Protein Diffraction Patterns

In X-ray Free Electron Laser (XFEL) experiments, protein molecules are shot with an X-ray beam and the scattering from the protein is recorded in a diffraction pattern as illustrated in Figure 1. The resulting diffraction patterns are unique to each protein and contain 2D information on its 3D structure. Because the orientation of the proteins cannot be controlled during the experiments, each 2D diffraction pattern corresponds to a particular view of the protein, and patterns are different even for the same protein 3D structure. In addition, proteins are flexible. As such, differences in diffraction patterns can also indicate a change in protein conformation. Therefore, to extract 3D information on protein structure, data processing and analysis of these low resolution 2D diffraction patterns is required. In addition, we should note that the intensity of the XFEL beam directly affects the quality of the images. Indeed, the signal to noise ratio decreases with a lower intensity beam thereby affecting the quality (or amount of information) of the image. Figure 2 shows three examples of resolutions produced by varying intensities of the XFEL beam (i.e., high, medium, and low). Specifically, in the figures, we represent high, medium, and low beam intensities as 1×10^{16} photons/μm^2/pulse, 1×10^{15} photons/μm^2/pulse, and 1×10^{14} photons/μm^2/pulse respectively.

B. Structural Properties

XFEL experiments can capture useful information of the structure and dynamics of biomolecules which are essential to elucidate their function. In this work, we are interested in annotating diffraction patterns. Several factors have to be

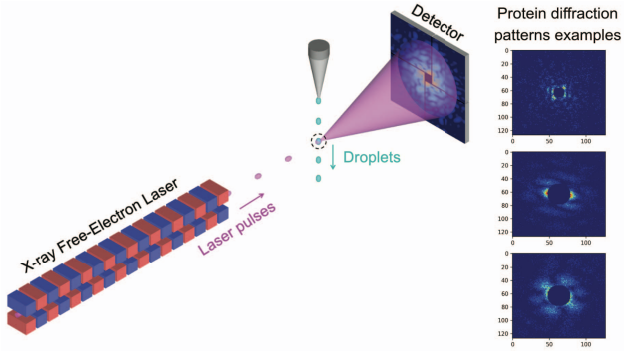


Fig. 1: Experimental process to obtain diffraction patterns (i.e., images) from 3D proteins using the X-ray Free Electron Laser (XFEL). Three examples of generated diffraction patterns are shown on the right.

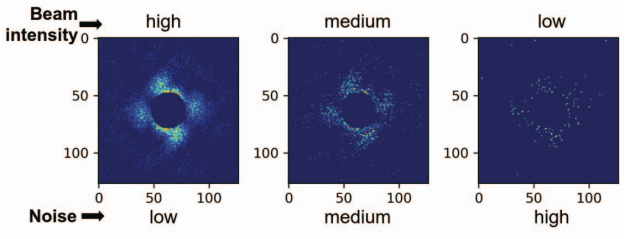
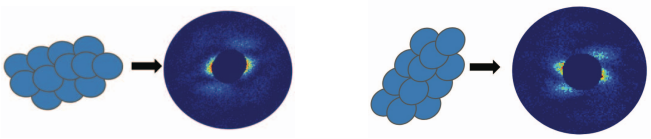


Fig. 2: Three resolutions of images created from three varying intensities of the incident beam (i.e., high, medium, and low) and with associated noise level (i.e., low, medium, and high).

considered: i) orientation of the protein captured in a given diffraction pattern, ii) the diversity in the protein structure (conformation), and iii) the type of protein.

Orientation refers to the placement of the incident beam with respect to a protein structure. It is described by the three Euler angles: Φ (Azimuth) = [-180,180], Θ (Altitude) = [0, 180], and Ψ (Psi or rotation angles = [0, 360] [4]. Figure 3 shows two orientations of the same protein, each with their own three angles (Φ, Θ, and Ψ). The diffraction patterns



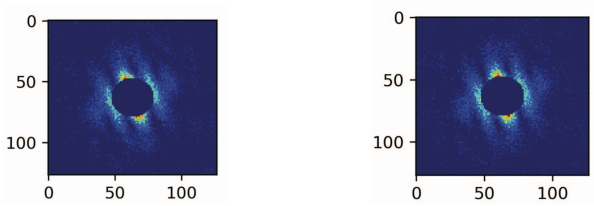
(a) Orientation 1: $\Phi, \Theta, \Psi = 24, 151, 346$

(b) Orientation 2: $\Phi, \Theta, \Psi = 145, 128, 291$

Fig. 3: Two different orientations for the same protein (i.e., Orientation 1 and Orientation 2).

can hide non-trivial symmetries associated with the Psi (Ψ) rotation angle. Observed symmetries are due to the physics associated with the diffraction process. For example, Figure 4 shows the same protein with two identical Φ and Θ angles but different Ψ rotation angle (i.e., 45 degree and 225 degrees

respectively). The two images look very similar despite the different rotation angle, exposing the structural similarities that may be present in the protein conformations.

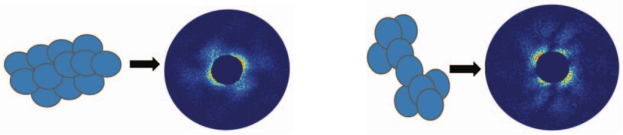


(a) Protein A; Conformation A; Φ, Θ,$\mathbf{\Psi}$ = 179, 64,**45**

(b) Protein A; Conformation A; Φ, Θ,$\mathbf{\Psi}$ = 179, 64,**225**

Fig. 4: Two diffraction patterns from the same protein type, conformation, and first two angles (Φ and Θ) with two different rotational angles (Ψ) that expose symmetry not present in the protein structure.

Conformation is the shape adopted by a protein and is caused by the rotation of the protein atoms around one or more single bonds. The protein atoms can assume a large number of possible spatial arrangements. Thus, XFEL may generate diverse diffraction patterns. Figure 5 shows two conformations (Conformation A and Conformation B) for the same protein.



(a) Conformation A; Φ, Θ,Ψ = 24, 151, 346

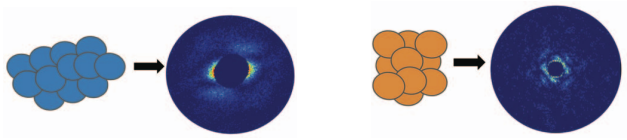
(b) Conformation B; Φ, Θ,Ψ = 34, 139, 106

Fig. 5: Two conformations for the same protein (i.e., Conformation A and Conformation B) generating two different diffraction patterns.

Protein type refers to the type of amino acids composing a protein. The 20 amino acids can be combined in different ways to make a protein. Different proteins are composed by different types and numbers of amino acids, and thus can assume different conformation structures that result in different diffraction patterns. Figure 6 shows two different proteins (i.e., Protein A and Protein B) with their corresponding conformation and orientation properties.

III. Protein Diffraction Identifier Framework

One contribution of this paper is the design and implementation of the XPSI framework to identify the structural properties of proteins (i.e., orientation, conformation, and protein type). The framework comprises four key modules: (i) a data pre-processing module to load, crop, and normalize images; (ii) an autoencoder module to extract diffraction pattern features; (iii) a ML module to model and predict the three structural properties from the diffraction patterns; and (iv) a validation module to measure the accuracy of the predicted properties. Figure 7 presents the XPSI framework and its modules.



(a) Protein A; Conformation A; Φ, Θ,Ψ = 24, 151, 346

(b) Protein B; Conformation C; Φ, Θ,Ψ = 84, 32, 82

Fig. 6: Two different proteins with different conformations and orientations (i.e., Protein type A and Protein type B).

A. Data Pre-processing Module

The input to our framework is a set of diffraction patterns (or images) that embed protein properties. The pre-processing module loads the dataset generated either experimentally (i.e., using XFEL beams) or from simulations (i.e., by downloading the proteins from PDB and using software such as Xmipp [5] and spsim [6] to recreate realistic diffraction patterns). Independently from the source of the diffraction patterns, the module crops the images to a fixed size. This is needed because not all the images in a dataset are the same size. Furthermore, the module deals with the defined beam intensity by performing a min-max normalization for which each pixel in an image is mapped to either black or white. This is needed to emphasize the embedded pattern. When generated experimentally, the properties of the considered proteins are not known a priori, and thus, the diffraction patterns are not annotated with their orientation and conformation. The type of protein is normally known. In this case, the pre-processing module forwards the dataset to the prediction module in which a model has been previously generated using, for example, simulated datasets. Images from datasets generated through simulations come annotated with the structural properties. The pre-processing module separates the images from the properties, converts the properties into metadata (i.e., labels), and appends the labels to a file with the properties for the entire dataset. The pre-processed dataset is forwarded to both the function extraction module and the modeling module for the model generation.

B. Feature Extraction Module

The pre-processed images are input to an autoencoder comprising an encoder and a decoder. The encoder maps an image into a tensor of dimension N, referred to as the latent space. The decoder rebuilds the image from the latent space. In our framework, the encoder architecture has three convolutional layers each followed by a max pooling layer, generating a matrix. A flatten layer transforms the matrix into a 1D array. The flatten layer is followed by a dense layer to

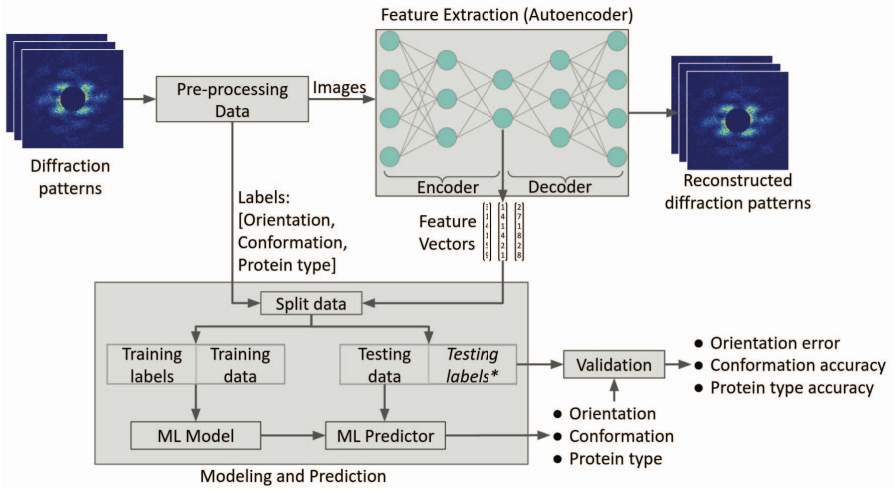


Fig. 7: The XPSI framework includes four modules: (i) a pre-processing module where the images are loaded, normalized, and cropped; (ii) a feature extraction module where by using an autoencoder the images are compressed and represented in a feature vector; (iii) a ML module where a KNN model is trained and tested to predict structural properties from pattern datasets; and (iv) a validation of the predictions. **The testing dataset for prediction does not require labels, but may contain them which can be used for the model validation.*

obtain the feature vector. The decoder has the reverse structure of the encoder. Figure 8 presents the encoder architecture. A

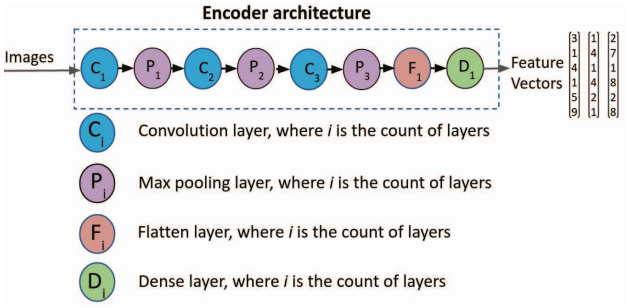


Fig. 8: The encoder architecture is composed of convolution, pooling, flatten and dense layers. The decoder has the reverse structure of the encoder.

key aspect of this architecture is that the dimension of the latent space has to be sufficient to faithfully reconstruct the original image. The closer the reconstructed image is to the original, the better the autoencoder is representing the image in a compressed format (i.e., latent space). This module allows users to select a latent space whose size is a trade-off between the architecture's accuracy and training time. To this end, the module measures the average mean squared error (MSE) between the reconstructed and the original images for different latent space sizes. Users can select a suitable size with the support of methods such as the elbow method, identifying when variance of the error and the associated gain in accuracy are not significant.

C. Modeling and Prediction Module

This module performs two distinct tasks: the generation of a model and the prediction of protein properties.

For the *modeling*, the module takes two inputs from a simulated dataset: the labels defining the protein properties (i.e., orientation, conformation, and protein type) from the pre-processing module and the feature vectors from the feature extraction module. Data is split into training and testing sets that are used for the modeling and prediction, respectively. The training dataset always includes labels. The testing dataset for prediction does not require labels, but may contain them which can be used for the model validation. Because the orientation consists of three continuous values, while the conformation and protein type consist of discrete values, the modeling solves one regression and two classification problems. We select a non-parametric algorithm, KNN, as the foundation of our modeling because of its reported high accuracy in both classification and regression problems and low execution costs. Comparisons with more expensive methods such as Random Forest are presented in Sec. V and support this design decision. As a result, we use a KNN-angle regressor for modeling the prediction of the angles of the orientation and two KNN-classificators for modeling the prediction of the different conformations and protein types, respectively. The modeling executes these three KNN models for each $K \in [2, M]$, with an incremental step and a maximum value of K (M). The M value and incremental step are user-defined parameters. An analysis of the root mean square error (RMSE) of the degree allows the module to identify the most suitable K number of neighbors. We use the randomized train-test split technique for estimating the performance of KNN on unseen data (i.e., data not used to train the model) [7]. To obtain a

more robust validation, we run and validate each model for a user-defined number of iterations. The final KNN model with the selected K is the output passed to the prediction.

For the *predictions*, we use the KNN models with the selected K and predict the three structural properties in either the testing dataset, if the dataset was generated with simulations, or the experimentally generated datasets otherwise. These predictions are the final output of the *Modeling and Prediction* module of our framework.

D. Validation Module

The validation module measures the accuracy of our framework predictions for datasets generated with simulations. To this end, the module uses two sets of metrics. For orientation accuracy, which is represented in the three Euler angles (Φ, Θ, Ψ), the module measures the error degree (ED) for the first two angles and Psi difference (PD) for the third. The error degree is defined as the distance between two points on a sphere given Φ (Azimuth) and Θ (Altitud), presented in Eq. 1. The point (Φ_2, Θ_2) represents the ground truth value and (Φ_1, Θ_1) the predicted one.

$$\sqrt{\sin^2(\frac{\theta_2 - \theta_1}{2}) + \cos(\theta_1)\cos(\theta_2)\sin^2(\frac{\phi_2 - \phi_1}{2})} \quad (1)$$

The Psi difference is defined in Eq. 2 as the difference between real and predicted Psi (Ψ) angle. When predicting the orientation (Φ, Θ, Ψ), the lower the error degree and the Psi difference, the more accurate XPSI is predicting the Φ, Θ, Ψ angles.

$$\psi_{real} - \psi_{predicted} \quad (2)$$

For conformation and protein type predictions, the module uses the accuracy metric in Eq. 3 that represents the ratio of correct predictions (both true positives and true negatives) over the total number of cases examined (i.e., true positives, true negatives, false positives, and false negatives). When predicting the conformation and the protein type, the higher the accuracy, the better XPSI predicts these two properties.

$$\frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

IV. Framework Accuracy for Different Scenarios

Using our framework, scientists can plug in their dataset and predict the dataset properties to answer their research questions. To demonstrate the versatility of our framework, we quantify and evaluate its ability to provide accurate predictions when presented with challenging datasets and combinations of structural protein properties.

A. Data: Simulated Diffraction Patterns

We use two datasets from protein structures of elongation factor 2 (EF2) and Ribosome. The datasets were generated through simulations so that we can leverage the annotations for the validation. Two conformations of EF2 with PDB ID 1n0u [8] and 1n0v [8] were downloaded from from Protein Data Bank [9] and used for the simulations. Two intermediate structures between 1n0u and 1n0v were generated using iterative normal mode analysis [10]. We used the two conformations Ribosome (PDB IDs 4KJ9/4KJA and 4KJB/4KJC) for the calculations [11]. Protein structures were rotated using Xmipp [5] to simulate different beam orientations, and the simulated XFEL diffraction patterns were generated using spsim [6].

Furthermore, the images in the datasets were generated with simulated different beam intensities: high (1×10^{16} photons/μm^2/pulse), medium (1×10^{15} photons/μm^2/pulse), and low (1×10^{14} photons/μm^2/pulse) for EF2; and low (1×10^{14} photons/μm^2/pulse) and lower (1×10^{13} photons/μm^2/pulse) for Ribosome. In our work, the beam intensity serves as a proxy for noise in images: the lower the beam intensity, the more noise appears in the pattern. There are 39,692 diffraction patterns per each conformation and beam intensity for all datasets. Table I summarizes the datasets. The diversity in the images in these datasets is representative of the complexity of predicting the structural properties of a protein.

TABLE I: Characterization of the simulated dataset including two protein types and multiple conformations, orientations, and beam intensities.

Protein type	Conformation	Orientation	Beam intensity
EF2	1n0u, mov20, mov53, 1n0vc	[Φ, Θ,Ψ]	high (1e16) medium (1e15) low (1e14)
Ribosome	9a, bc	[Φ, Θ,Ψ]	low (1e14) lower (1e13)

B. Setup: Evaluation Scenarios

To demonstrate the robustness and versatility of our framework, we design three evaluation scenarios by selecting challenging combinations of properties and prediction tasks. *Scenario 1* challenges our framework to identify the *orientation* of the proteins, even in the presence of symmetric patterns that can obfuscate the prediction. *Scenario 2* is designed to assess whether our framework can differentiate between *conformations* with very similar, but not identical, structures of the same protein. Finally, *Scenario 3* targets our framework's capability to identify the *protein identity* by merging diverse datasets of diffraction patterns with multiple orientations, conformations, and protein types. We further increase the complexity of the predictions by using images with different beam intensities. A summary of the structural properties present in each scenario is included in Table II.

We use the same methodology for each scenario: we randomly split the data in 90% to generate the model (training) and 10% to predict the protein properties (testing). For the search of K in the model, we set the maximum value of K (M) equal to 20 and an increment step equal to 1; we conduct 10 iterations per K. The prediction accuracy per trial is measured on the testing dataset. Table III summarizes the outcome for the three scenarios that are described in detail below.

TABLE II: Summary of the evaluation scenarios. Each scenario uses a selection of datasets from Table I and includes the targeted structural property.

Scenario	Main Target	Protein Type	Conformation	Beam Intensity
1	Orientation	EF2	1n0u, 1n0vc	high, medium, low
2	Conformation	EF2	mov20, mov53	high, medium, low
3	Protein identity	EF2	1n0u, 1n0vc	high, low
		Ribosome	9a, bc	low

TABLE III: Validation results for the scenarios defined in Table II. For each scenario there are different beam intensities for which the selected K is listed. Columns 4 and 5 are the validation metrics for orientation which indicate the percentile of the testing dataset within that error. Columns 6 and 7 show the percentage of accuracy when predicting conformation and protein type respectively. NA=Not Applicable.

Scenario	Beam Intensity	K Selection	Error Degree (% of Data)	Psi Difference (% of Data)	Conformation Accuracy	Protein Type Accuracy
1	high	K=2	$< 10(95\%)$	$< 10(95\%)$	100%	NA
	medium	K=8	$< 10(75\%)$	$< 12(75\%)$	99%	
	low	K=2	$< 65(75\%)$	$< 35(75\%)$	92%	
2	high	K=2	$< 10(95\%)$	$< 10(95\%)$	97%	NA
	medium	K=2	$< 10(75\%)$	$< 10(75\%)$	90%	
	low	K=4	$< 45(75\%)$	$< 35(75\%)$	80%	
3	EF2 low, Ribosome low	K=2	$< 45(75\%)$	$< 10(75\%)$	86.5%	100%
	EF2 high, Ribosome low	K=4	$< 10(95\%)$	$< 12(95\%)$	90%	100%

C. Scenario 1: Predicting Orientation in Symmetric Patterns

Identifying the orientation of a protein is challenging when the diffraction pattern presents some degree of symmetry. The symmetry in diffraction patterns is not caused by the symmetry in the protein structure. Diffraction patterns are approximately symmetric for any protein due to the physics behind the diffraction process, as shown in Figure 4. In order to evaluate the robustness of our framework to differentiate a conformation and its 180-degree rotation, we use the EF2 dataset in Table I; the dataset includes symmetrical diffraction patterns. We use our framework to identify the orientation with rotation of the protein type EF2 for two conformations (i.e., 1n0u and 1n0vc) and three beam intensities (i.e., high, medium, and low).

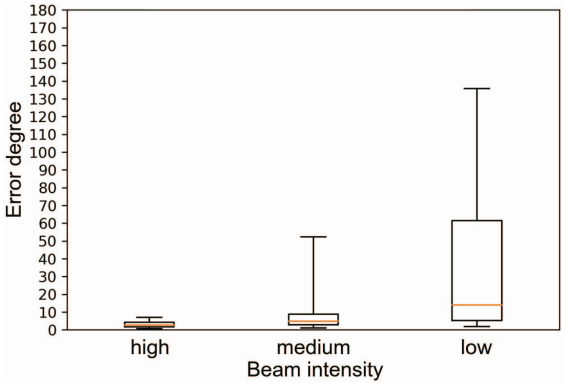
The first row of Table III shows the validation results of XPSI in Scenario 1. The framework validates the predictions of the angles through measuring the error degree in Eq. 1 for the first two angles, and the Psi difference in Eq. 2 for the rotation angle. Table III shows the average error degree and Psi difference of the testing dataset over the 10 iterations. We visualize the error degree for the two first angles in Figure 9a and the difference of the predicted versus the actual value of the Psi angle in Figure 9b. The box-plots display the 5th percentile, 25th percentile, 50th percentile (orange bar also known as median), 75th percentile, and 95th percentile of the testing dataset (i.e., 10% of the input data randomly selected). For the high beam intensity dataset, XPSI predicts the first two angles with an error degree equal or below 10° for 95th percentile of the testing dataset and the Psi angle with a difference of 10° for the 95th percentile of the dataset. The predictions are very accurate when the beam intensity is high, but they degrade as the patterns become noisier. However, our framework is still able to provide an orientation estimation when the rotation angle exposes symmetry in the diffraction patterns. Low beam intensity dataset is predicted with an error degree below 60° for the 75th percentile of the testing dataset and the rotation angle with a difference of 35° for the 75th percentile of the testing dataset. This result shows how our framework can estimate approximate angles even for diffraction patterns with very low signals.

In this scenario, we also differentiate between the 1n0u and 1n0vc conformations. We observe that for high, medium, and low beam intensity the framework identifies the conformation with 100%, 99%, and 92% accuracy, respectively. The protein type accuracy does not apply in this scenario since we have only one protein type (EF2).

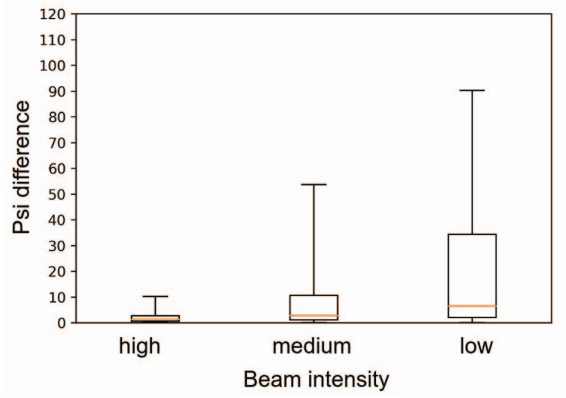
D. Scenario 2: Distinguishing Similar Conformations

The goal of this scenario is to demonstrate that our framework is able to distinguish conformations even when they have similar structures. Note that two conformations with very similar structures yield diffraction patterns that are very similar. We select the EF2 protein because we can generate conformations with similar but not identical structures such as mov20 and mov53. In Figure 10 we show the intermediate structures of the protein type EF2, where mov20 and mov53 have almost indistinguishable conformations to the naked eye. We generate a dataset of diffraction patterns with the three beam intensities (high, medium, and low).

The second row of Table II shows the validation results for Scenario 2. Our framework correctly identifies the conformations with 97%, 90%, and 80% accuracy for the high, medium and low beam intensities, respectively. Figure 11 shows the confusion matrices for the three beam intensities. The higher the intensity of the beam, the higher the accuracy and the less incorrectly classified patterns we have. Our framework is able to correctly predict the conformation in 72.8% samples even in the presence of the noise introduced by low intensity beams. These results support that our framework is able to



(a) Error degree.



(b) Psi difference.

Fig. 9: Orientation error for the three beam intensities (high, medium, and low) in Scenario 1 for two EF2 protein conformations (1n0u and 1n0vc). The box-plot displays the 5th percentile, 25th percentile, 50th percentile (orange bar also known as median), 75th percentile, and 95th percentile of the testing dataset (i.e., 10% of the input data randomly selected).

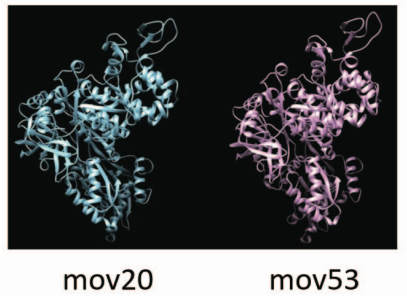
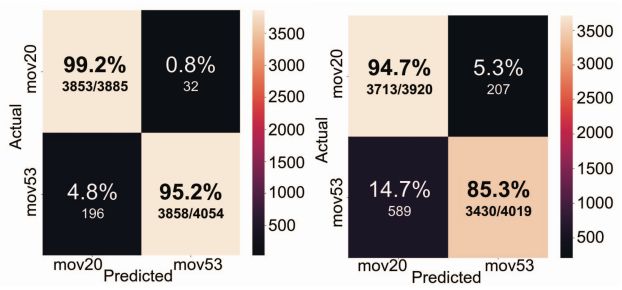


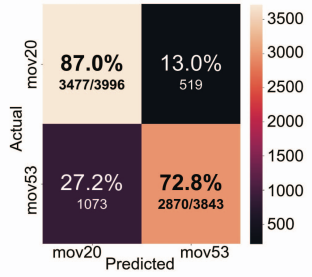
Fig. 10: The intermediate conformations of the EF2 protein. The conformations mov20 and mov53 have a similar structure but are not identical.

differentiate between conformations in challenging scenarios, including almost indistinguishable conformations and noisy data.



(a) High

(b) Medium



(c) Low

Fig. 11: Confusion matrices for high, medium, and low beam intensity when differentiating between mov20 and mov53 conformations. The confusion matrix indicates the number of samples that were correctly predicted and the number that were not.

E. Scenario 3: Identifying Proteins and Their Properties

The main goal of this scenario is to evaluate the capability of our framework to identify protein identity in a general and realistic set of diffraction patterns with multiple conformations and orientations. This dataset includes diffraction patterns from EF2 and Ribosome in two conformations each, and with diverse orientations. We include low beam intensity datasets for EF2 and Ribosome to make sure there is a level of noise both proteins share.

The fourth row of Table II shows a 100% of protein type identification accuracy in Scenario 3. This remarkable result is accompanied by very high conformation accuracy. Figure 12 shows the confusion matrix of the conformation predictions. We observe that XPSI predicts 1n0u, 1n0vc, 9a, and bc with an accuracy of 88.3%, 85.2%, 92.3%, and 80.2%, respectively. When predicting across all four conformations, our framework delivers an average of 86.5% of conformation accuracy. It is important to highlight that for both cases the inaccurate prediction of the conformation is made within the protein type, which indicates that there are no cases in which the framework misidentifies conformations from different protein types. These results show how our framework is able to distinguish between conformations from different protein types accurately. This facilitates the understanding of the complex conformational dynamics even when the conformations are from different protein types.

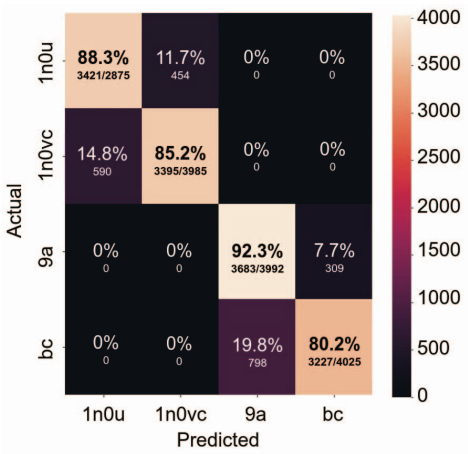
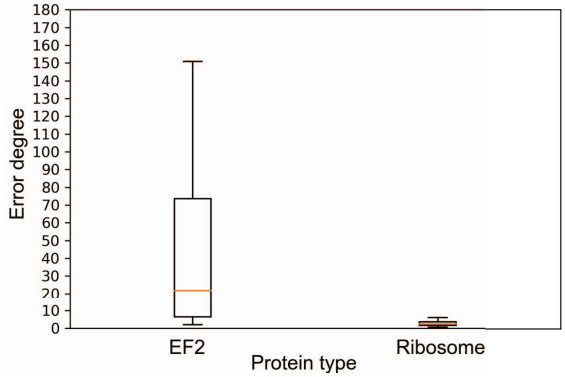
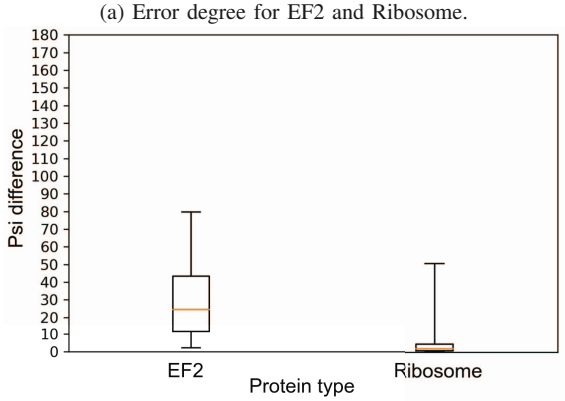


Fig. 12: Confusion matrix for Scenario 3 with differentiating between 1n0u, 1n0vc, 9a and bc conformations. The first two conformations belong to the EF2 protein while 9a and bc belong to Ribosome. We use the low beam intensity datasets for both protein types. The confusion matrix indicates the number of samples that were correctly predicted and the number that were not.



(a) Error degree for EF2 and Ribosome.



(b) Psi difference for EF2 and Ribosome.

Fig. 13: Orientation error for Scenario 3 given each protein type: EF2 and Ribosome. The box-plot displays the 5th percentile, 25th percentile, 50th percentile (orange bar also known as median), 75th percentile, and 95th percentile of the testing dataset for each protein type (i.e., 10% of the input data randomly selected).

The framework predicts the orientation for the diffraction patterns of both protein types (EF2 and Ribosome). We visualize the error degree and Psi difference for each protein type in the testing dataset in Figure 13. The box-plots display the 5th percentile, 25th percentile, 50th percentile (orange bar also known as median), 75th percentile, and 95th percentile of the testing dataset for each protein type (i.e., 10% of the input data randomly selected). We observe in Figure 13a that XPSI predicts the first two angles with an error degree equal or below 70° and 10° for the 75th percentile of the EF2 testing dataset and the 95th percentile of the Ribosome testing dataset, respectively. In Figure 13b we present the Psi difference. XPSI predicts the Psi angle with a difference equal or below 40° and 6° for the 75th percentile of the EF2 and Ribosome testing dataset, respectively. The Ribosome molecule is larger in size than EF2 (≈20 times). Therefore, it is more accurate the prediction of orientation for Ribosome when both protein types datasets have low beam intensity.

Since the EF2 is significantly smaller than ribosome, using the diffraction patterns with stronger beam intensity, we can imitate the situation where two proteins have similar size. That is why, we test for the mixture of the diffraction patterns of EF2 with high beam intensity and Ribosome with low beam intensity. We obtain the same protein type accuracy of a 100%. Additionally, in Figure 14 the conformation prediction accuracy increases compared to the low beam intensity case for both protein types. Three conformations (1n0u, 1n0vc, and 9a) are now predicted with an accuracy greater than 93.5%. Also, XPSI predicts the first two angles with an error degree equal or below 10° for 95th percentile of the testing data combining EF2 with high beam intensity and Ribosome with low beam intensity. The Psi angle is predicted with a difference equal or below 5° for 95th percentile of the testing data combining EF2 with high beam intensity and Ribosome with low beam intensity.

Our evaluation demonstrates that our framework (i) identifies rotation in the diffraction patterns, even in the presence of symmetry; (ii) differentiates between conformations with similar, but not identical, structures of the same protein; and (iii) identifies the type of a protein in a completely diverse set of diffraction patterns with multiple conformations and orientations. All of these capabilities are proven with different beam intensities. As expected, the lower the beam intensity the noisier the diffraction patterns, which affects the accuracy of the predictions.

V. Comparisons with Other Methods

We compare our framework in terms of its computational costs and prediction accuracy to other existing ML methods:

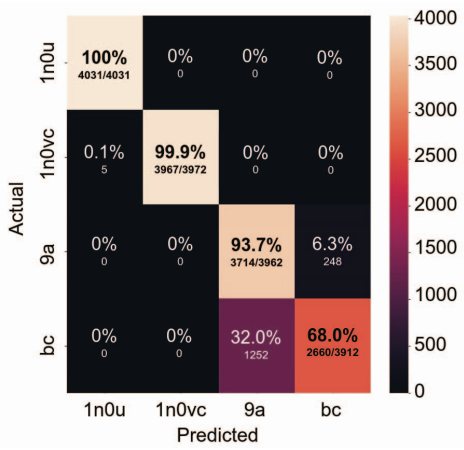


Fig. 14: Confusion matrix for Scenario 3 when differentiating between 1n0u, 1n0vc, 9a and bc conformations. The first two conformations belong to the EF2 protein type while 9a and bc belong to Ribosome. We use the high beam intensity dataset for EF2 and low beam intensity dataset for Ribosome.

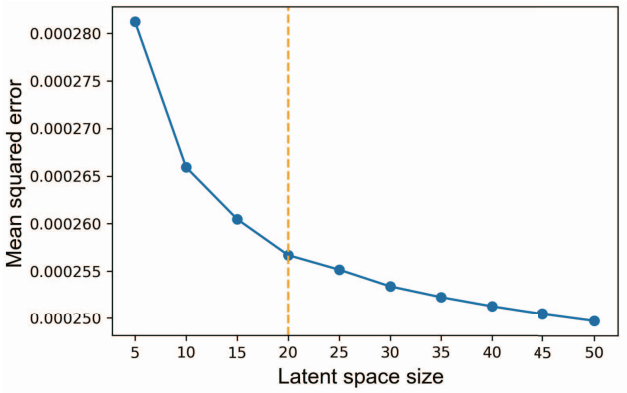


Fig. 15: Image reconstruction error for different sizes of the latent space in the feature extraction module presented in Sec III-B. Based on the elbow method, we select 20 as our size of the latent space.

Random Forest (RF) [12], [13] and Neural Networks (NN). We select Scenario 1 (EF2 high beam intensity dataset with two conformations and their rotational orientations) as comparison benchmark since it allows us to perform regression and classification on the same dataset. This allows us to conduct a fair comparison of the computational cost and prediction accuracy of the different methods. We execute all our tests on the same platform, a single node of an IBM Power9 cluster with 128 GB RAM and 1 NVIDIA Volta V100 GPU.

A. Cost of Feature Extraction for XPSI and RF

We compare the regression capability of the KNN method in our framework to another broadly used method such as RF. Both KNN and RF use feature vectors that we generate with our feature extraction module (see Sec. III-B). In this module we train our autoencoder for different sizes of the latent space (LS) $LS \in [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$ for 100 epochs with an adamax optimizer and a batch size of 16. Figure 15 shows the mean squared error (MSE) for each latent space; using the elbow method to select the latent space. Based on the figure, we select a latent space of size 20; after a latent space of 20 the variance of the error and the associated gain in accuracy are not significant. It takes 15h 27m to run these 10 autoencoders with different sizes of latent space, for an average of 1h 32m for each autoencoder. This feature extraction module is a one-time cost: the off-line process runs once and the selected autoencoder with its latent space is used by the modeling and prediction module in Sec. III-C. This time is presented on Row 4 of Table IV.

B. Cost of Modeling and Prediction with XPSI and RF

Both KNN and RF are regression methods. Contrary to KNN that predicts angles using neighbors information, RF predicts angles based on the combination of multiple decision trees. The RF method is widely used in different applications because of its high level of interpretability. We generate a RF-based module performing both the modeling and prediction similarly to our framework. In other words, we replace our original modeling and prediction module in XPSI with the RF-based module.

We feed the RF-based module with the labels defining the protein properties (i.e., orientation, conformation, and protein type) from our pre-processing module and the feature vectors from our feature extraction module. As in XPSI, we split the input dataset in 90% for training and 10% for testing. The RF-based module builds a forest with a certain number T of decision trees that are chosen from a user-defined list of values. An analysis of the RMSE of the error degree allows the RF-based module to identify the most suitable number of decision trees to become the final RF model used for predictions. The RF model is used to predict the orientations in the testing dataset we used in Scenario 1. For both XPSI and the RF-based modeling, we conduct 10 iterations for each value of K and T respectively, where K is the number of neighbor in our KNN and T is the number of decision trees in RF. We measure the modeling time using KNN and RF in Scenario 1 (see Table IV).

In the table we observe that, for the modeling, it takes $4m$ to generate the KNN-regressor model with $K \in [2, 10]$ and an incremental step of 1, while it takes $10h22m$ to build five forests using RF with $T \in [10, 50, 100, 500, 1000]$. We consider a larger number of decision trees in RF than the number of neighbors in KNN to reduce any bias in the sampled data used to generate forests. For the prediction, we measure the time that takes the KNN model with $K = 2$ and the RF model with $T = 1000$ to predict the orientation of the testing dataset. These two values are selected because of their lowest RMSE. KNN takes $1.37s$ to predict the orientations in the testing dataset while RF takes $2.45s$. For the prediction accuracy, we observe that our KNN model predicts the two

TABLE IV: Accuracy and computational performance comparison of XPSI, Random Forest (RF) and Evolutionary Neural Architecture Search (NSGA-NET). ED: Error Degree, PD: Psi Difference, and CA: Conformation Accuracy.

Problem Type	Regression		Classification	
Structural Property	Orientation		Conformation	
Method	XPSI	RF	XPSI	NSGA-NET
Off-line process	15 h 27 m		15 h 27 m	20 h
Modeling Time	4 m	10 h 22 m	4.4 m	20.75 m
Prediction Time	1.37 s	2.45 s	1.65 s	1.90 s
Prediction Accuracy	ED=$< 10(95\%)$ PD=$< 10(95\%)$	ED=$< 73(95\%)$ PD=$< 48(95\%)$	CA=100%	CA=100%

first angles (Φ and Θ) with an error degree (ED) below 10° for the 95th percentile of the testing dataset, while RF does it with a much larger error of 73°. The same applies for the Psi difference (PD), where our model predicts the third angle (Ψ) with a difference below 10° for the 95th percentile of the testing dataset, while RF does it with a bigger difference of 48°. Overall, we observe that the RF-based module takes a large amount of time for both modeling and prediction while delivering less accuracy than our modeling and prediction module.

C. Cost of Modeling and Prediction with XPSI and NNs

We compare the classification capability of our framework to another broadly used ML method such as Neural Networks (NNs). Specifically, we measure time and accuracy to extract features, generate a model, and predict protein conformations with XPSI versus time and accuracy to execute a neural architecture search (NAS) and use the best NN to model and predict the same property.

Neural networks are a popular method for classification, especially for image datasets. Finding suitable NNs is a time-consuming process involving several rounds of hyperparameter selection, training, validation, and manual inspection [14]. To address this challenge, we select the Non-dominated Sorting Genetic Algorithm for the Neural Network Architecture Search (NSGA-NET) [15]. NSGA-NET conducts a neural architecture search that automates the design of NN models using multi-objective optimization (MOO). In this case, we instruct NSGA-NET to generate NN models that maximize classification accuracy while minimizing FLOPS during the prediction stage for the purpose of maximizing computational efficiency. NSGA-NET is evolutionary, and we run it for 10 generations with population and offspring size of 10. Thus, at the end of the NAS we obtain 100 models that each train and validate for 10 epochs. We choose an NN model that balances conformation accuracy and computational efficiency, and we train it to completion for 20 epochs.

Table IV shows the results of running XPSI and NSGA-NET on Scenario 1 for classification of protein conformations. In Row 4 of Table IV, we compare the one-time cost of executing the feature extraction module with the one-time cost of the neural architecture search (NAS). It takes $15h27m$ to extract the features for XPSI while it takes NSGA-NET $20h$ to train and validate different NNs in their search. Furthermore, we observe that it takes $4.4m$ to generate the KNN-classificator model with $K \in [2, 10]$ and an incremental step of 1 while it takes $20.75m$ to train the selected NN model for 20 epochs. For the prediction, we measure the time that takes XPSI and NSGA-NET to classify the conformations in the testing dataset. XPSI takes $1.65s$ while the NN model takes $1.9s$. Both XPSI and the NN models are able to classify protein conformation of the testing data with 100% accuracy. However, the modeling time is shorter when using our framework and its KNN model.

VI. Conclusions

We design, implement, and evaluate XPSI (X-ray Free Electron Laser-based Protein Structure Identifier), a framework capable of predicting three structural properties of proteins (i.e., orientation, conformation, and protein type) from their diffraction patterns. Our framework predicts orientation despite symmetry in the diffraction patterns for the EF2 protein with an error degree and Psi difference within 10° for the 95th percentile of the testing dataset (unseen data) for high beam intensity. Also, it predicts conformation with an accuracy of 97% even when the structural shapes of the conformations are nearly indistinguishable. In addition, our framework identifies between two protein types (EF2 and ribosome) with 100% in a general and realistic set of diffraction patterns with multiple conformations and orientations. The addition of imaging factors in our experimental setup, such as beam intensity, symmetric diffraction patterns, increases the complexity of our data and prediction task. Therefore, employing new techniques such as VAEs (Variational autoencoders) or end-to-end deep learning frameworks are promising next steps for our framework. As future work, we can improve prediction of the orientation using a tiered approach, where first we identify the protein type (the most accurate prediction we have), then filter by conformation, and finally refine the orientation prediction.

Acknowledgment

This work is supported by NSF awards #1741057, #1841758 and #2028923, #2138811, and #2223704, in collaboration with the JLESC Consortium, the JDRD Program at UTK, the IBM Shared University (SUR) Award, the JSPS KAKENHI JP20H05453 award, and the FOCUS for Establishing Supercomputing Center of Excellence. The authors acknowledge Piotr Luszczek, Michael Wyatt, and Neil Lindquist for their support in the development of the framework.

Available Jupyter Notebooks

The Jupyter Notebook implementing the XPSI framework can be found at: https://github.com/TauferLab/XPSI.

REFERENCES

[1] N.-T. D. Loh and V. Elser, "Reconstruction algorithm for single-particle diffraction imaging experiments," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 80, p. 026705, aug 2009.

[2] M. Tegze and G. Bortel, "Atomic structure of a single large biomolecule from diffraction patterns of random orientations," *J Struct Biol*, vol. 179, pp. 41–45, may 2012.

[3] M. Nakano, O. Miyashita, S. Jonic, C. Song, D. Nam, Y. Joti, and F. Tama, "Three-dimensional reconstruction for coherent diffraction patterns obtained by XFEL," *Journal of synchrotron radiation*, vol. 24, pp. 727–737, Jul 2017.

[4] J. B. Heymann, M. Chagoyen, and D. M. Belnap, "Common conventions for interchange and archiving of three-dimensional electron microscopy information in structural biology," *J Struct Biol*, vol. 151, pp. 196–207, aug 2005.

[5] J. M. de la Rosa-Trevín, J. Otón, R. Marabini, A. Zaldívar, J. Vargas, J. M. Carazo, and C. O. S. Sorzano, "Xmipp 3.0: an improved software suite for image processing in electron microscopy," *J Struct Biol*, vol. 184, pp. 321–328, sep 2013.

[6] Filipe Maia, "Single particle diffraction simulator, spsim." [Online, 05-25-2022].

[7] A. Rácz, D. Bajusz, and K. Héberger, "Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification," *Molecules*, vol. 26, no. 4, 2021.

[8] R. Jørgensen, P. A. Ortiz, A. Carr-Schmid, P. Nissen, T. G. Kinzy, and G. R. Andersen, "Two crystal structures demonstrate large conformational changes in the eukaryotic ribosomal translocase," *Nature Structural & Molecular Biology*, vol. 10, pp. 379–385, may 2003.

[9] Berman, H M and Westbrook, J and Feng, Z and Gilliland, G and Bhat, T N and Weissig, H and Shindyalov, I N and Bourne, P E, "The Protein Data Bank," *Nucleic Acids Res*, vol. 28, pp. 235–242, jan 2000.

[10] O. Miyashita, J. N. Onuchic, and P. G. Wolynes, "Nonlinear elasticity, proteinquakes, and the energy landscapes of functional transitions in proteins," *Proceedings of the National Academy of Sciences*, vol. 340, no. 22, pp. 12570–12575, 2003.

[11] A. Pulk and J. H. D. Cate, "Control of ribosomal subunit rotation by elongation factor G," *Science (New York, N.Y.)*, vol. 340, pp. 1235970–1235970, jun 2013.

[12] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[13] L. Breiman, "Random Forests," in *Machine Learning*, vol. 45, pp. 5–32, 2001.

[14] A. Keller Rorabaugh, S. Caíno-Lores, T. Johnston, and M. Taufer, "Building High-Throughput Neural Architecture Search Workflows via a Decoupled Fitness Prediction Engine," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2913–2926, 2022.

[15] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, "NSGA-Net: Neural Architecture Search Using Multi-Objective Genetic Algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, (New York, NY, USA), p. 419–427, Association for Computing Machinery, 2019.